

Exhibit 2

Exhibit 2



White Paper - EAA

Ellipsus' Enterprise Application Adapters

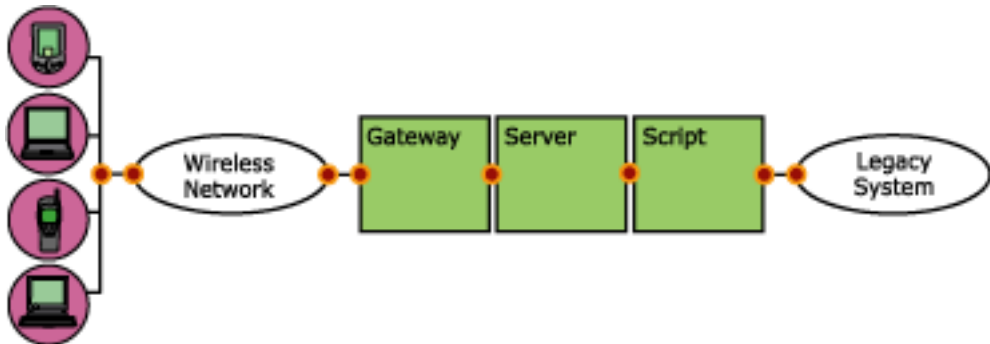
Mobile access to legacy systems will be business critical

Mobile Internet opens up new possibilities for enterprises, which have mobile work forces or mobile customers. Communication standards are in place that are the basis for boundary free movement. Devices, such as cellular phones, personal digital assistants and laptops are readily available, and the telecom networks are being adapted to packet communication (GPRS for example). Operators are now also offering mobile services. Businesses are therefore beginning to explore mobile Internet to see how it can improve business for them, their employees and their customers. This applies not only to the ISP/Telecom industry, but the enterprise in general. Banks, insurance companies, stock trading firms a s o, will all benefit from this technology. Mobile access to the legacy system will be business critical for many of these enterprises.

Preserve your prior investments

Most enterprises have made earlier investments in legacy systems. They already have computer networks, hardware and operating system platforms, applications and a substantial amount of information stored in databases.

It is a challenge to integrate mobile access with the legacy system and still be able to preserve earlier investments, preserve system integrity, maintain existing functionality and add new functionality.



Common technology

Common technology suggests the use of server scripts, or some dynamic server engine, for integration, this has several drawbacks. One of those drawbacks is the issue of maintaining state (information about how a service progresses for each user). This could be solved by using cookies, but this results in unnecessary use of memory and communication overhead, which is costly, degrades performance and reduces the battery life in mobile devices.

Certain goals must be met in order to ensure successful integration with mobile technology:

Straightforward and unambiguous

The communication channel should be straightforward and unambiguous. If the legacy system uses object-oriented interfaces, these interfaces should be visible all the way to the client device. This makes the system easy to understand and easy to adapt as new requirements emerge.

Keep the amount of data to a minimum

The solution should keep the amount of communicated data to a minimum. This promotes performance, longer battery life and economical operation.

Separate logical structure from content

The solution should separate logical structure from presentation and content. This makes it possible to accommodate a wide variety of client devices. Changes, to content, are also less likely to affect the logic of the application.

Ellipsus' Enterprise Application Adapters - EAA

The Enterprise Application Adapters of the infiniteMASS™ mobile application server from Ellipsus enables mobile access to legacy systems through its adapter infrastructure. This infrastructure allows systems integrators/developers to use common technologies for distributed computing in a new exciting fashion.

Supports industry standards

infiniteMASS supports industry standards for object-oriented distributed computing, such as the Common Object Request Broker Architecture (CORBA), Java Remote Method Invocation (RMI), Distributed Component Object Model (DCOM); and Simple Object Access Protocol (SOAP). With infiniteMASS the legacy system can expose objects all the way to the mobile client. The object interface can have any shape or form. The openness of this interface is an extremely powerful tool for mobile integration.

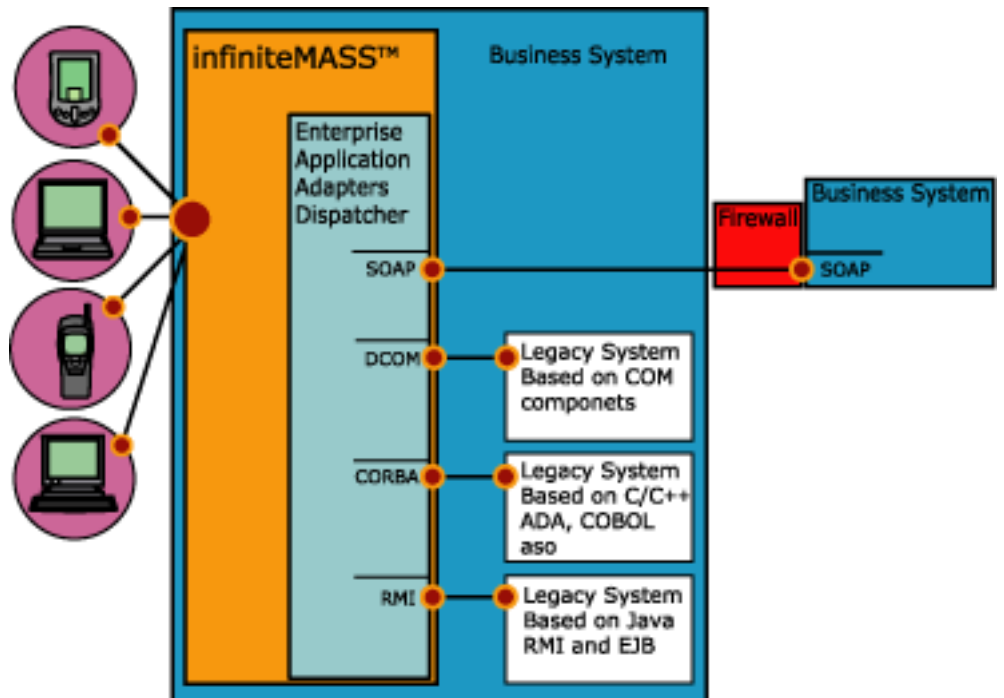
Create flexible and dynamic interfaces

It is possible to create flexible and dynamic interfaces, which optimize communication and offer high performance. The mobile user will therefore be delivered responsive and user-friendly services.

infiniteMASS supports four technologies today. They can be used in parallel and combined with other web systems, such as server pages and scripts.

To choose suitable technology is easy and straight forward:

Technology	Should be used
Java RMI	when the legacy system is developed in Java™ in a distributed environment. Enterprise Java Beans (EJB) and EJB servers can exist.
CORBA	when the legacy system is distributed and developed in C/C++, ADA, COBOL or some other programming language, or any combination of the above. CORBA is language independent and therefore perfect for integration with legacy systems.
DCOM	in business-to-business (B2B) scenarios, in order to connect COM Components.
SOAP	in business-to-business (B2B) scenarios, in order to connect legacy systems from remote offices or business partners. A big benefit with SOAP is that it uses the normal http connection and can therefore work through firewalls without any specific configurations.



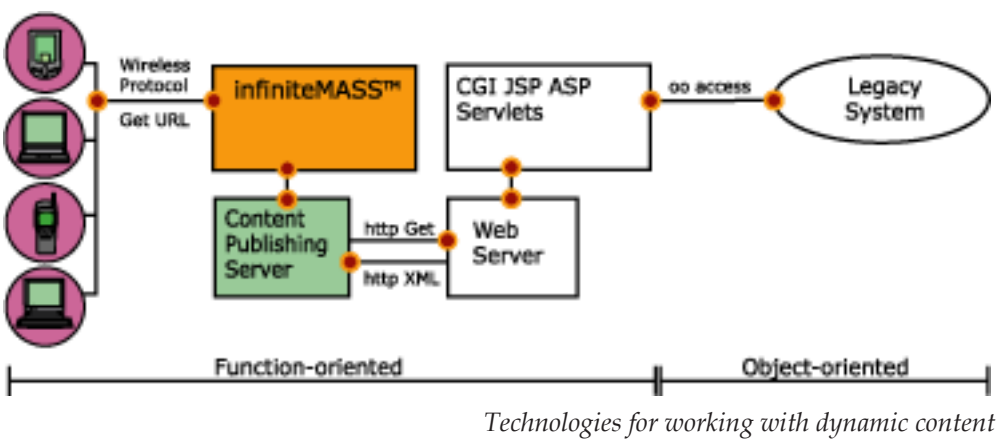
Each technology can be viewed as a channel for access to the legacy system.

infiniteMASS will dispatch messages to the appropriate channel (RMI, CORBA, DCOM or SOAP) based on the structure of the incoming request from the mobile device. This request is in the form of a Unified Resource Locator (URL), which will typically indicate channel, object, method and parameters.

An exemple

For example, if the legacy system has a function for calculating salaries, the URL for this could be <http://hostname/rmi/salaryObject/calculate?start=0&stop=10000>. This request is using RMI to communicate with a remote Java object, which calculates the salaries for employees with employee numbers 0 to 10.000. These types of URLs are normally hidden behind a menu displayed on the client device. While one menu item points to RMI another may very well point to CORBA, DCOM or SOAP. The underlying technology is transparent to the device, yet the object-oriented technology, and thus the system design, is visible all the way from the legacy system to the device. This is a prerequisite for separating logic with presentation and content. Keeping them separate is important in order to reach the largest user community possible in a cost efficient way.

A number of technologies are available for working with dynamic content. Examples include Common Gateway Interface (CGI) scripts, Java Server Pages (JSP), Active Server Pages (ASP) and Servlets. Ellipsus' products can work with all of them. A typical scenario is illustrated below.



A client (the mobile user) will issue a request for information via a wireless protocol, such as WAP. infiniteMASS will extract the binary content from the protocol data and decode it into standard format (HTTP). The request is forwarded to the Content Publishing Server, which will connect to the appropriate web server. The web server will interface to a dynamic technology (CGI, JSP, ASP or Servlets), which, in turn, will access the legacy system.

This scenario raises a number of issues that the developer/integrator needs to address:

How to manage state

Dynamic web applications must maintain state for each client. For example: in a ticket booking system, the client decides where to go and when to travel, how to pay a s o. The state is communicated back and forth between the client and the server as cookies, often combined with temporary web pages, created for the specific session. This adversely affects overall performance since it involves access to the file system and more communication.

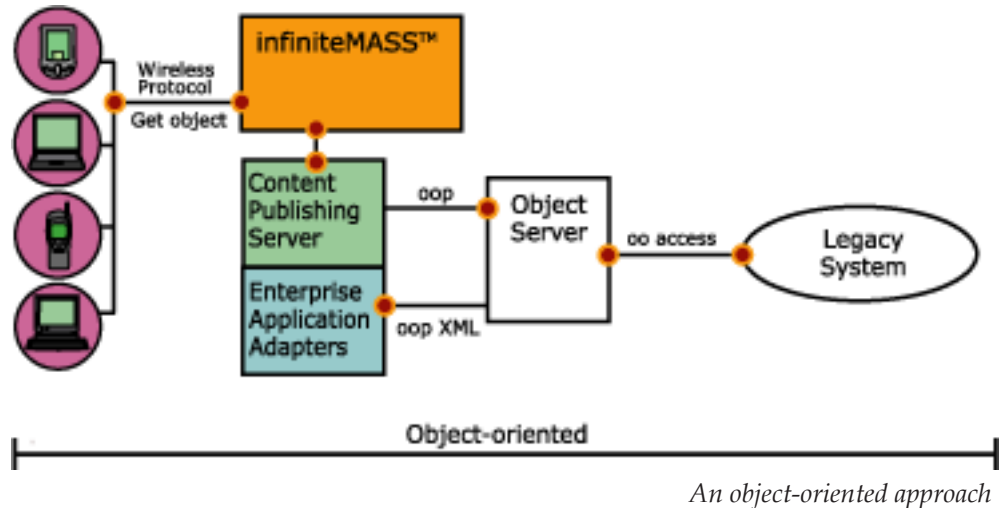
Function-oriented interfaces

Most systems today use object-oriented technology. While a servlet, for example, can access the legacy system in an object-oriented fashion, the web server must use a function-oriented interface to the servlet. The object model is therefore lost in the communication chain. This makes it impossible to define access independent interfaces to the legacy system.

Integration

The developer/integrator is still left with the actual integration issue, since this is not a natural component of these technologies. The result is yet another integration module, which needs to be called from the server page, CGI script or servlet.

An alternative approach is to use infiniteMASS which is an object-oriented approach to integrating the mobile Internet with business systems.



A client (the mobile user) will again issue a request for information via a wireless protocol, such as WAP. But, rather than specifying a URL pointing to content, the URL will specify an object, a method in that object, and any parameters to the method. infiniteMASS will extract the binary content from the protocol data and decode it into standard format (HTTP). The request is forwarded to the Content Publishing Server, which will connect to the appropriate object server using the Enterprise Application Adapters module. The object server will return a reference to a distributed object (Java RMI, CORBA or SOAP), which will access the legacy system directly.

This scenario resolves the issues as follows:

How to manage state

State is a natural part of objects. Since each session can be associated with its own object, that object can maintain and update its state as required. There is no longer any need to communicate the state back and forth between the server and the client. The need to use secondary storage is also removed since the object is maintained in random access memory for the duration of the session. The result is better performance due to less communication and less file management.

Function-oriented interfaces

The Enterprise Application Adapters functions of infiniteMASS makes the communication chain object-oriented from source (legacy system) to destination (client device). The client device will specify object-method-parameters rather than web page. There are no additional requirements on the client devices in order to make them object-oriented.

Integration

This is what the infiniteMASS Enterprise Application Adapters functionality is all about. The legacy system can expose its objects to the mobile client without any intermediate components.

The Enterprise Application Adapters functions of infiniteMASS is of great interest to systems integrators and developers since it is based on industry standards. The uniqueness lies in the way the system is implemented. The distributed objects can have any interface, reflecting the logic of the legacy system rather than a required interface to the Enterprise Application Adapters.

This leads to a number of benefits when compared to alternative approaches:

<i>Fast and more responsive</i>	Faster, more responsive applications, since the state is maintained in the objects, on the server side, which improves overall performance.
<i>Longer endurance</i>	Client devices with longer endurance, due to less communication and therefore also less battery usage.
<i>Easier to accomodate</i>	Easier to accommodate different device types, due to a natural separation of logic from content and presentation.
<i>More efficient development</i>	More efficient system development, since the object-oriented approach promotes reusability.
<i>Easier to model</i>	Easier to model the complete system, since the object-oriented model works all the way to the client device.
<i>Access to legacy systems</i>	Access independent interfaces to the legacy system, which promotes reuse and consistency.
<i>Ease of integration</i>	Ease of integration with legacy systems, since Java RMI, CORBA and SOAP are supported. Some of these technologies allow mixing of programming languages as well as mixing function-oriented and object-oriented architectures.

About Ellipsus System

Ellipsus Systems develops, markets, and supports a suite of universal, platform neutral middleware products that dynamically provision and manage applications and content to a wide range of consumer mobile device types. Through its highly regarded mobile application provisioning technology and infinite suite of software products, Ellipsus enables mobile operators and enterprises to extend their reach to the mobile Internet by leveraging existing infrastructure with secure, scalable and open solutions. Founded in 1999, Ellipsus Systems, Inc. is a closely-held Delaware corporation with corporate and North American headquarters in Dallas, Texas, and sales and engineering offices in Sweden.



www.ellipsus.com • info@ellipsus.com • wap.ellipsus.com

Corporate Headquarters, Dallas: Ellipsus Systems Inc • P.O. Box 797525 • Dallas, TX 75379-7525, USA
Phone: +1-214-228-6252

Nordic Sales Office, Stockholm: Ellipsus Systems AB • Ringvägen 100 C • SE-118 60 Stockholm, Sweden
Phone: +46 (0)8-556 97 330 • Fax: +46 (0)8-556 97 339

Engineering Center, Växjö: Ellipsus Systems AB • Regementsgatan 9 • SE-352 36 Växjö, Sweden
Phone: +46 (0)470-73 71 60 • Fax: +46 (0)470-73 71 61